BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Parallel Data Storage, Analysis, and Visualization of a Trillion Particles

Suren Byna, J. Chou, O. Rübel, Prabhat, H. Karimabadi, W. S. Daughton, V. Roytershteynz, E. W. Bethel, M. Howison, K.-J. Hsu, K.-W. Lin, A. Shoshani, A. Uselton, and K. Wu

Lawrence Berkeley National Laboratory

Tsinghua University, Taiwan

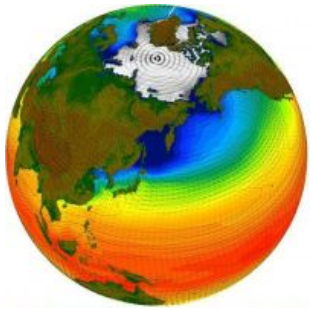
University of California - San Diego

Los Alamos National Laboratory

Brown University

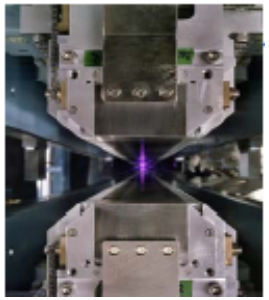
Data Explosion in Scientific Computing

- **Modern scientific discoveries are driven by data**



By 2020, climate data is expected to be hundreds of exabytes or more

LHC experiments produce petabytes of data per year



Light source experiments at LCLS, ALS, SNS, etc. produce tens of TB/day

1 Exabyte per a day (10 petabytes every hour)

- Storing, analyzing, and visualizing large data are big challenges

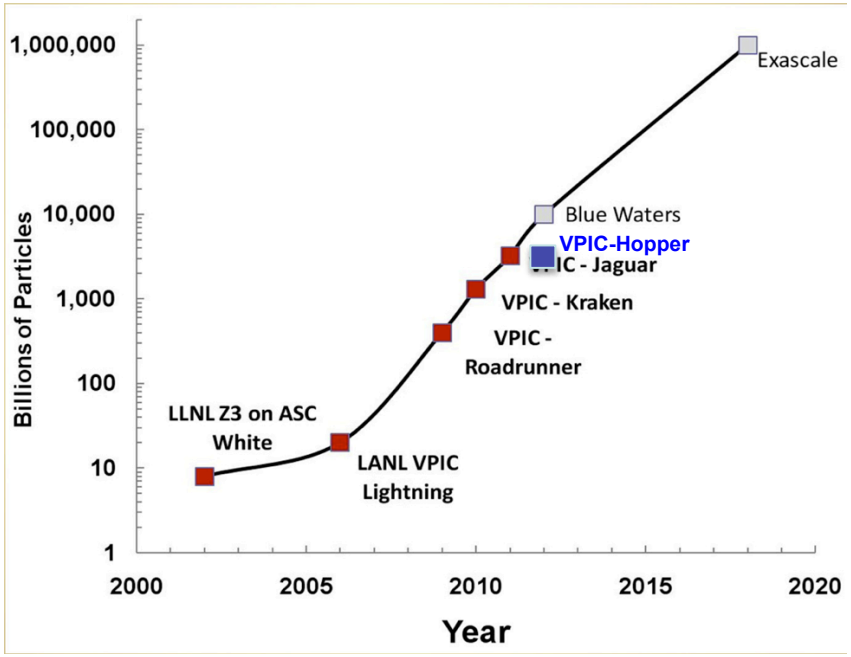
Data Management Challenges

- A scalable I/O strategy for storing massive data output
  - In situ analysis works well when analysis tasks are known *a priori*
  - Many scientific applications require to store datasets for exploratory analysis
- A scalable strategy for conducting analysis on these datasets
  - Sift through large amounts of data looking for useful information
- A visualization strategy for examining the datasets
  - Display information that makes sense

- VPIC is a simulation that pushes the limits of data management tools on large supercomputers

Vector Particle-in-Cell (VPIC) Simulation

- A state-of-the-art 3D electromagnetic relativistic PIC plasma physics simulation
- It is an exascale problem and scales well on large systems
- An open boundary VPIC simulation of magnetic reconnection
- NERSC Hopper Supercomputer
  - 6,384 compute nodes; 2 twelve-core AMD 'MagnyCours' 2.1-GHz processors per node; 32 GB DDR3 1333-MHz memory per node; Interconnect with a 3D torus topology

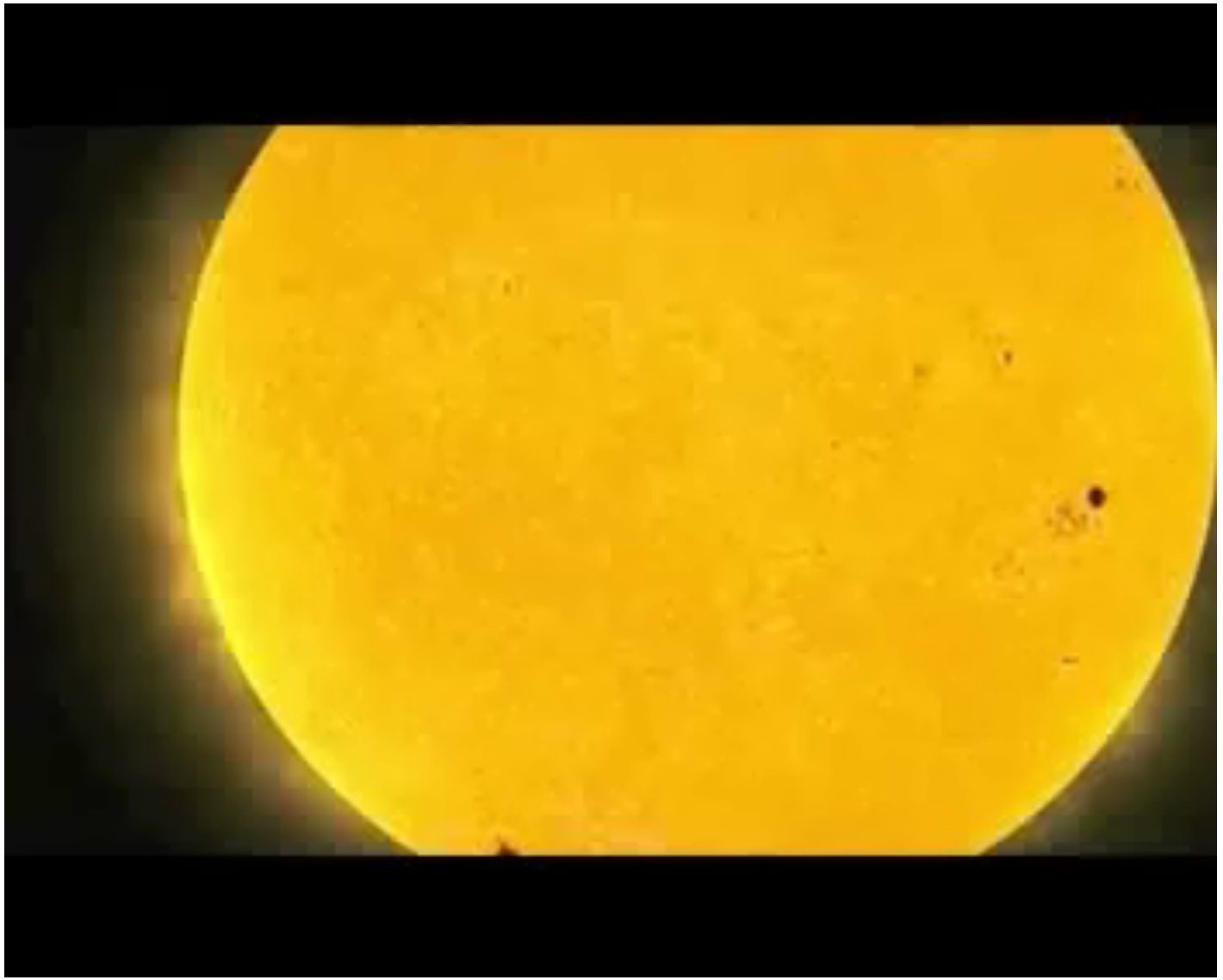


BERKELEY LAB

Trillion Particle Simulation

- Simulates collisionless magnetic reconnection
  - Reconnection is an important mechanism that releases energy explosively as field lines break and reconnect in plasmas
  - Example: Earth's magnetosphere's reaction to solar eruptions
- Trillion electrons and trillion ions
  - Uses 120,000 CPU cores
  - 20,000 MPI processes and each process has 6 OpenMP threads
  - Writes electron data of 30 to 42 TB every ~2000 time steps
  - Total 12 such particle datasets [380 TB]
  - Electric and Magnetic field data is another 100 TB

http://www.youtube.com/watch?v=i_x3s8ODaKg



Animation from NASA

Trillion Particle Simulation

- Simulates collisionless magnetic reconnection
  - Reconnection is an important mechanism that releases energy explosively as field lines break and reconnect in plasmas
  - Example: Earth's magnetosphere's reaction to solar eruptions
- Trillion electrons and trillion ions
  - Uses 120,000 CPU cores
  - 20,000 MPI processes and each process has 6 OpenMP threads
  - Writes electron data of 30 to 42 TB every ~2000 time steps
  - Total 12 such particle datasets [380 TB]
  - Electric and Magnetic field data is another 100 TB

Science Questions

- Understanding physical mechanisms responsible for producing magnetic reconnection in a collisionless plasma
- Analysis of highly energetic particles
  - Are the highly energetic particles preferentially accelerated along the magnetic field?
  - What is the spatial distribution of highly energetic particles?
- Agyrotropy: A quantitative measure of the deviation of the distribution from cylindrical symmetry about the magnetic field
- What are the properties of particles near the reconnection hot-spot (the so-called X-line)?
  - What is the degree of agyrotropy in the spatial vicinity of the X-line?

System Configuration

- NERSC Hopper supercomputer
  - 6,384 compute nodes
  - 2 twelve-core AMD 'MagnyCours' 2.1-GHz processors per node
  - 32 GB DDR3 1333-MHz memory per node
  - Employs the Gemini interconnect with a 3D torus topology
  - Lustre parallel file system with 156 OSTs at **a peak BW of 35 GB/s**
- Software
  - Cray's MPI library (xt-mpt 5.1.2)
  - HDF5 1.8.8 library with H5Part 1.6.6
  - FastQuery and FastBit
  - VisIt 2.4

Our Tools and Techniques

- Scalable I/O strategy for storing particle data
  - H5Part: A simple API on top of HDF5 (NOT **HDFS**) to read/write particle data
  - Performance tuning for Lustre striping optimizations
- Scalable strategy for conducting analysis on these datasets
  - FastBit: Bitmap index generation and querying software
  - Hybrid Parallel FastQuery
    - ✓ API to generate bitmap indexes
    - ✓ API to query indexed or data from different data formats (HDF5, NetCDF, and ADIOS-BP)
- Visualization strategy for examining the datasets
  - Query-driven visualization using VisIt

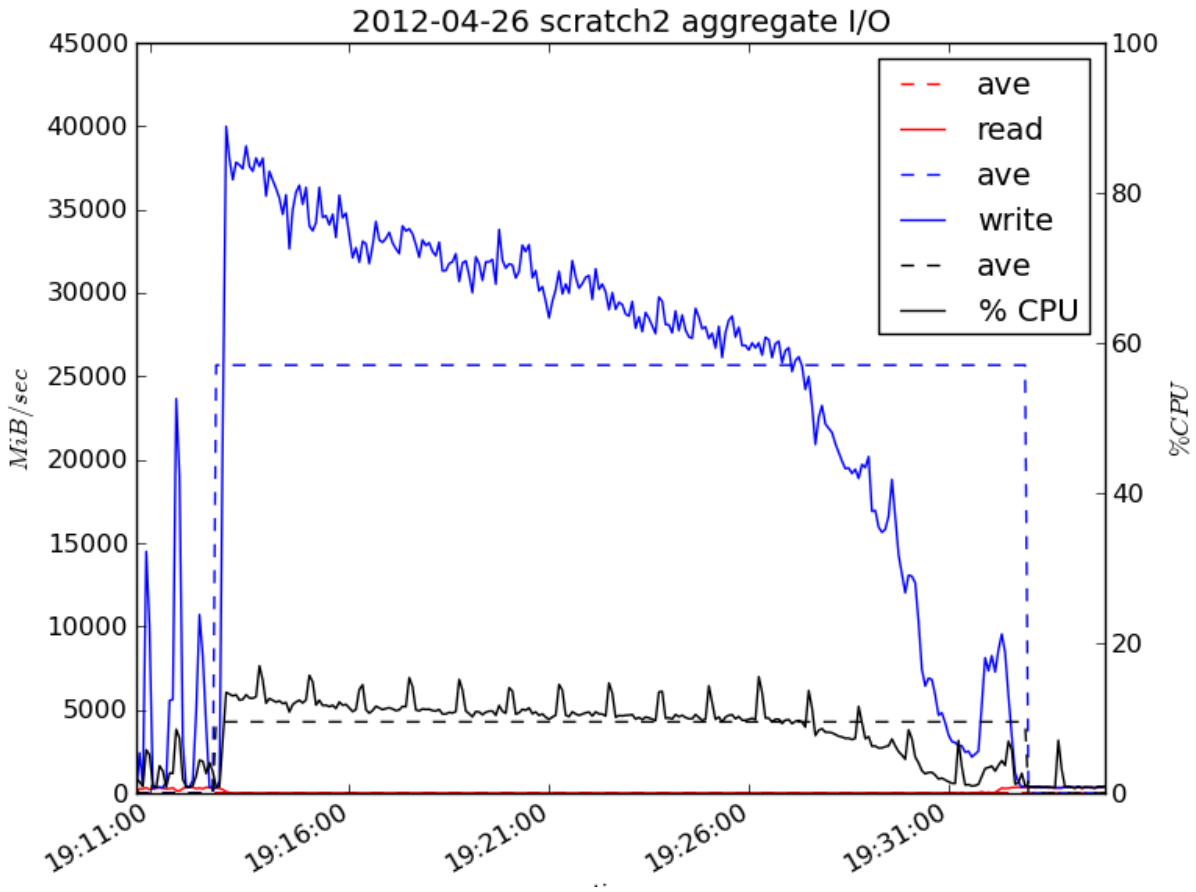
Our Tools and Techniques

- Scalable I/O strategy for storing particle data
  - H5Part: A simple API on top of HDF5 (Hierarchical Data Format, NOT **HDFS**) to read/write particle data
  - Performance tuning for Lustre striping optimizations
- Scalable strategy for conducting analysis on these datasets
  - FastBit: Bitmap index generation and querying software
  - Hybrid Parallel FastQuery
    - ✓ API to generate bitmap indexes
    - ✓ API to query indexed or data from different data formats (HDF5, NetCDF, and ADIOS-BP)
- Visualization strategy for examining the datasets
  - Query-driven visualization using VisIt

Writing Particle data

- File-per-process (FPP)
  - MPI-Domain level aggregation
  - Gating technique: Instead of all processes writing at the same time, a few processes were allowed to write data simultaneously
  - Data format: Flat binary
  - Problems
    - Data management and visualization tools only support standard formats such as HDF5, NetCDF, etc.
    - FPP model dictates the concurrency of subsequent stages in the analysis pipeline
- Single shared file with self-describing data format
  - Our choice: HDF5
  - If File-process-process is the measure of “good” performance, can single-shared-file approach achieve that?

I/O performance with file-per-process



- 20,000 MPI domains write one file per process
- Amortized I/O rate of 26 GB/s

Writing Particle data

- File-per-process (FPP)
  - MPI-Domain level aggregation
  - Gating technique: Instead of all processes writing at the same time, a few processes were allowed to write data simultaneously
  - Data format: Flat binary
  - Problems
    - Data management and visualization tools only support standard formats such as HDF5, NetCDF, etc.
    - FPP model dictates the concurrency of subsequent stages in the analysis pipeline
- Single shared file with self-describing data format
  - Our choice: HDF5
  - If File-process-process is the measure of “good” performance, can single-shared-file approach achieve that?

Writing Particle data

- File-per-process (FPP)
  - MPI-Domain level aggregation
  - Gating technique: Instead of all processes writing at the same time, a few processes were allowed to write data simultaneously
  - Data format: Flat binary
  - Problems
    - Data management and visualization tools only support standard formats such as HDF5, NetCDF, etc.
    - FPP model dictates the concurrency of subsequent stages in the analysis pipeline
- Single shared file with self-describing data format
  - Our choice: HDF5
  - If file-per-process is the measure of “good” performance, can single-shared-file approach achieve that?

Particle writing code in H5Part

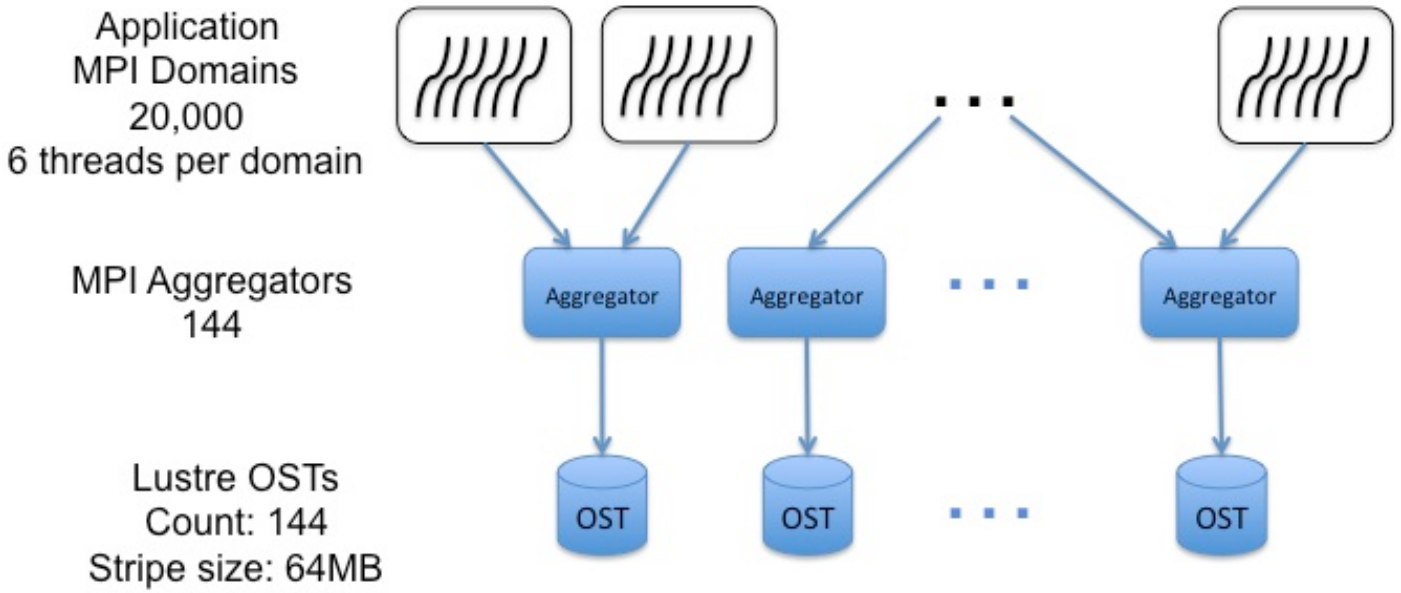
```
h5pf = H5PartOpenFileParallel (fname, H5PART_WRITE |
                    H5PART_FS_LUSTRE, MPI_COMM_WORLD);
H5PartSetStep (h5pf, step);
H5PartSetNumParticlesStrided (h5pf, np_local, 8);

H5PartWriteDataFloat32 (h5pf, "dX", Pf);
H5PartWriteDataFloat32 (h5pf, "dY", Pf+1);
H5PartWriteDataFloat32 (h5pf, "dZ", Pf+2);
H5PartWriteDataInt32   (h5pf, "i",  Pi+3);
H5PartWriteDataFloat32 (h5pf, "Ux", Pf+4);
H5PartWriteDataFloat32 (h5pf, "Uy", Pf+5);
H5PartWriteDataFloat32 (h5pf, "Uz", Pf+6);
H5PartWriteDataFloat32 (h5pf, "q", Pf+7);

H5PartCloseFile (h5pf);
```

http://vis.lbl.gov/Research/H5Part/

Parallel I/O Setup

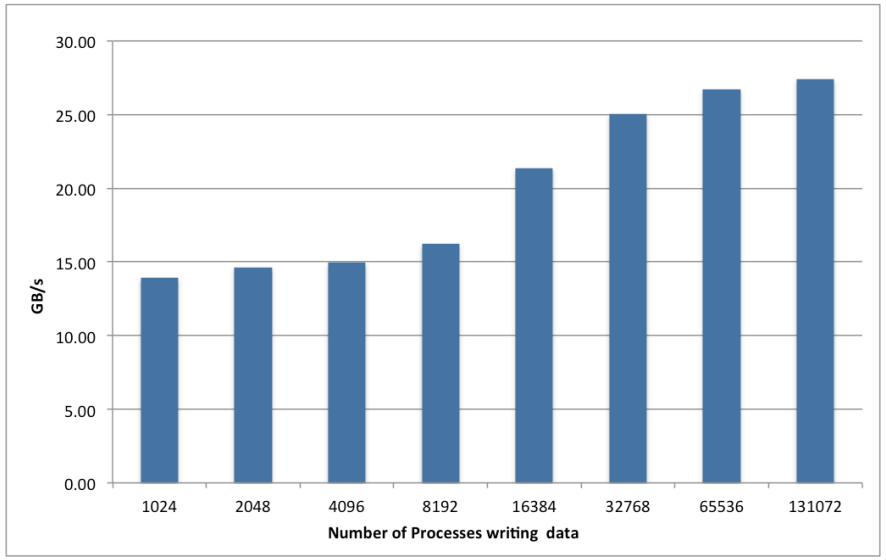


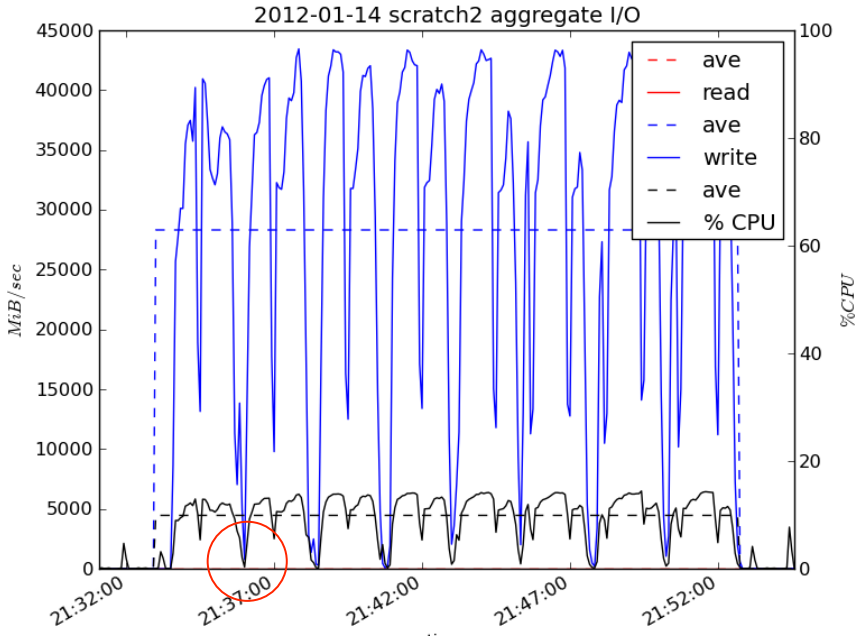
- Collective buffering
  - A subset of MPI tasks, called aggregators, collect data into a temporary buffer and write the data to I/O servers
- File Striping
  - Lustre stripes files across multiple disks (i.e. Object Storage Targets or OSTs)

Performance Tuning on Lustre file system

- Stripe count: The number of OSTs across which a file is written
- Stripe size: The number of bytes written on one OST before cycling to the next
- Lustre-aware MPI-IO implementation
  - MPI collective buffer size is equal to the stripe size
  - Number of MPI aggregators is equal to the stripe count
- Lustre Tuning
  - Varied stripe count from 64 to 156 and stripe size from 1MB to 1GB
  - Choice of 144 stripes and 64MB stripe size was optimal for writing particle data

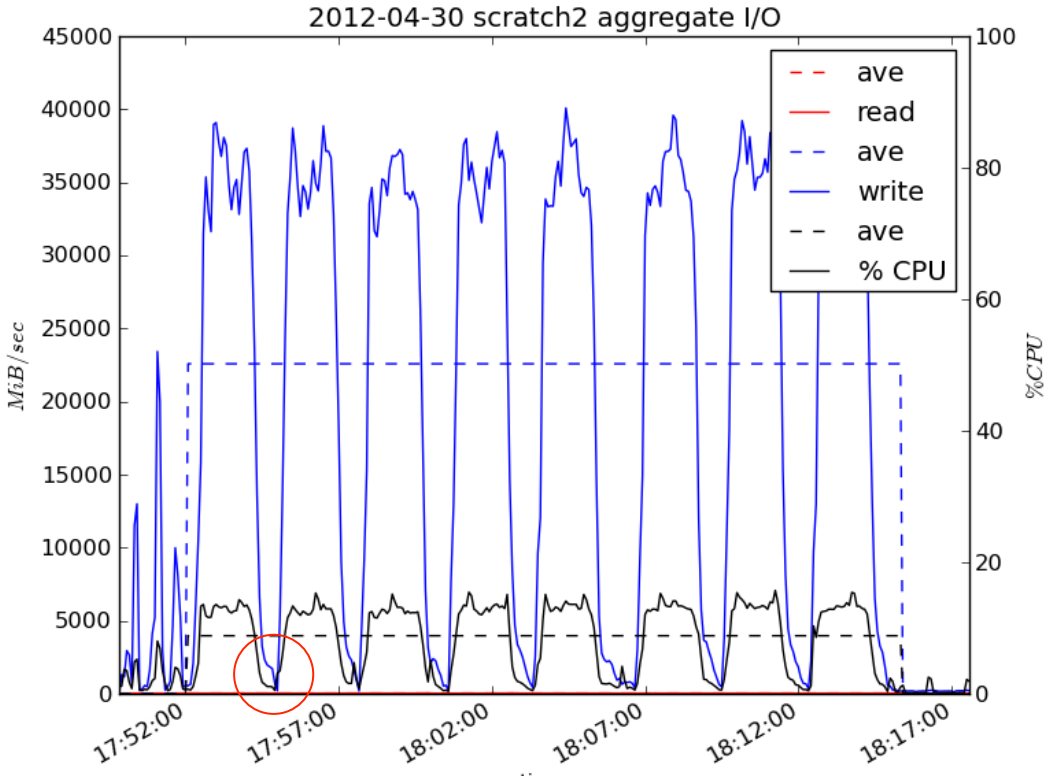
Performance of VPIC I/O Kernel





- Scaling study of increasing data with increasing number of processes
- Each process wrote 8M particles (uniform number for all MPI domains)
- At 128K cores, VPICBench writes 32TB data at an I/O rate of 27 GB/s
- Transient I/O rates using Lustre Monitoring Tool shows reaching peak I/O rates

Performance of Writing Trillion Particles



- Reached I/O peak rate in writing each variable
- Amortized I/O rate of 24 GB/
- Found 12 OSTs running 30% slower causing a slight slowdown
- Recent particle dumps without faulty OSTs show 27 GB/s rate

BERKELEY LAB

Our Tools and Techniques

- Scalable I/O strategy for storing particle data
  - H5Part: A simple API on top of HDF5 to read/write particle data
  - Performance tuning for Lustre striping optimizations
- Scalable strategy for conducting analysis on these datasets
  - FastBit: Bitmap index generation and querying software
  - Hybrid Parallel FastQuery
    - ✓ API to generate bitmap indexes
    - ✓ API to query indexed or data from different data formats (HDF5, NetCDF, and ADIOS-BP)
- Visualization strategy for examining the datasets
  - Query-driven visualization using VisIt

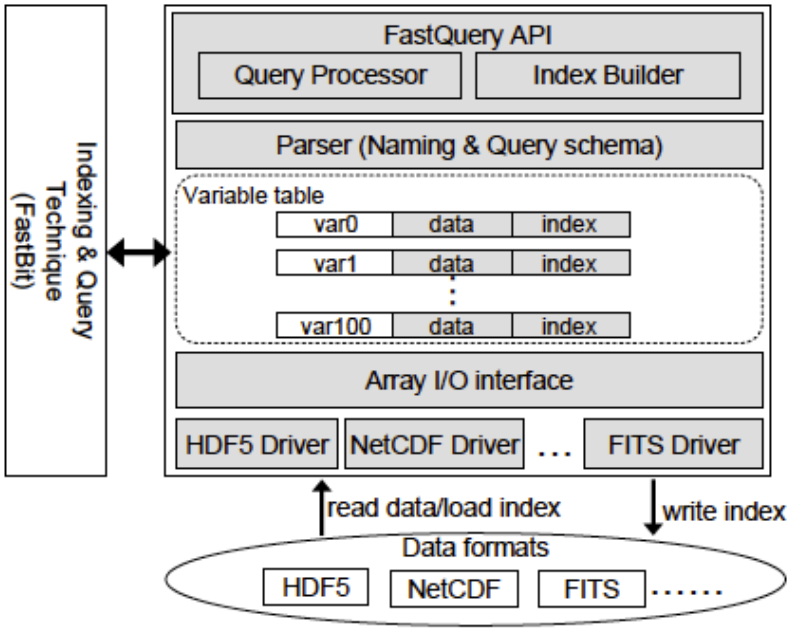
Data Analysis: Indexing and Querying

- Most data analysis tasks are performed on a subset of a dataset with interesting features
- To find that subset data quickly, database technology has SQL
- **FastBit** provides querying capability for fast scientific data access
  - Indexes and stores each column data separately
  - Generates compressed bitmap indexes
  - Search speeds are 10X-100X better than the best known bitmap indexing methods
  - Size: On average about 1/3 of data volume compared to 3-4 times in common indexes (e.g. B-trees)

https://sdm.lbl.gov/fastbit/

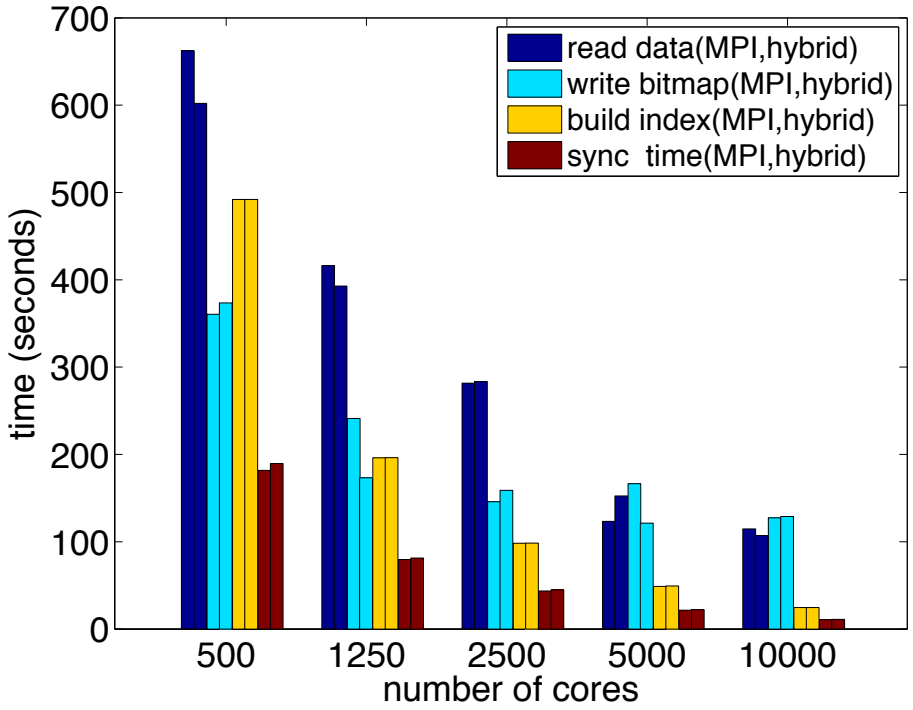
Hybrid Parallel FastQuery

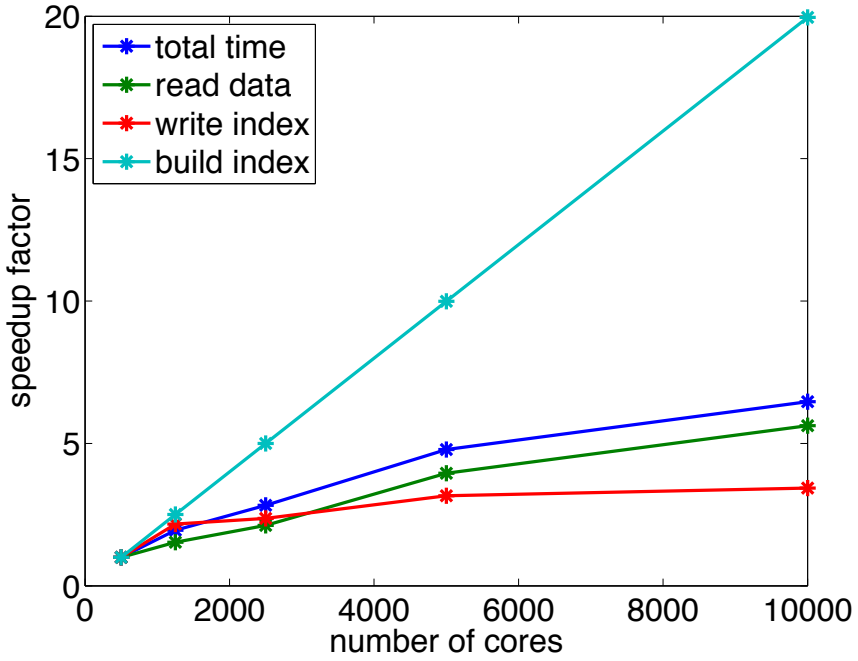
- FastQuery: An API to query arbitrary scientific data formats
- Uses FastBit indexing and querying technology
- Parallel implementation previously used MPI-only
- Recent implementation uses hybrid parallelism to utilize multicore CPUs
  - MPI tasks spawn a fixed number of threads (used Pthreads)
  - Threads perform indexing and querying functions on fixed size sub-arrays of data assigned to them



https://codeforge.lbl.gov/projects/fastquery/

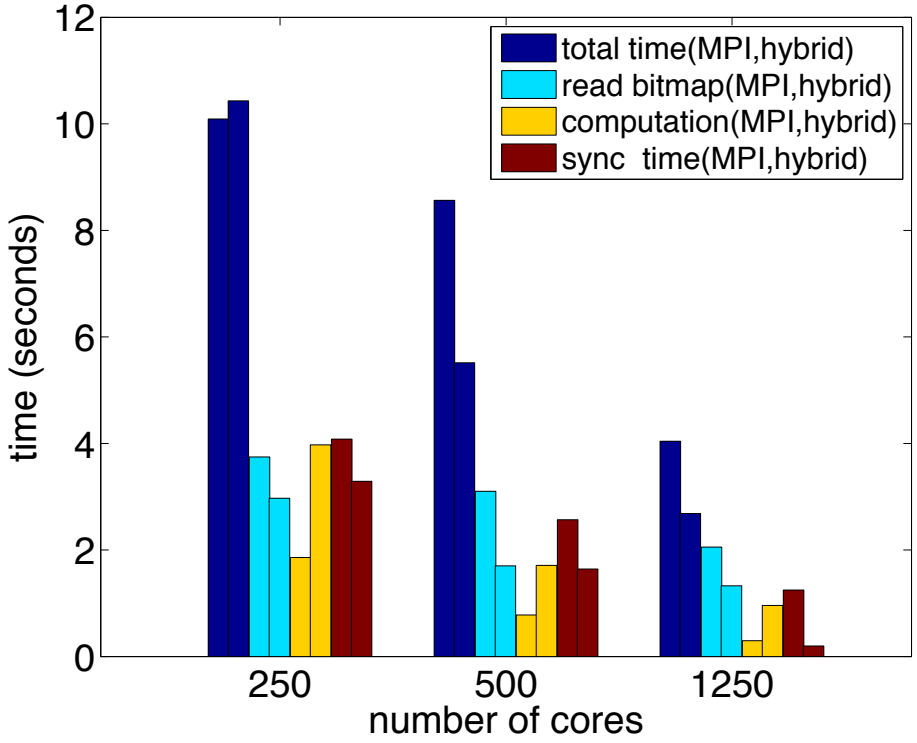
Performance of Indexing with Hybrid FastQuery





- Three steps: Read → Build Indexes → Write Indexes
- Overall advantage of hybrid parallel implementation over MPI-Only implementation
- Building index step scales linearly, but I/O doesn't scale

Performance of Querying with Hybrid FastQuery



#cores	scan	MPI-alone	hybrid
250	975	10.1	10.8
500	532	8.6	5.5
1250	266	4.1	2.7

- Queried for particles where 'Energy > 1.3' from the trillion dataset
- Took **less than three seconds** to sift through 1 trillion particles
- Better than MPI-only

Our Tools and Techniques

- Scalable I/O strategy for storing particle data
  - H5Part: A simple API on top of HDF5 to read/write particle data
  - Search for Lustre striping optimizations
- Scalable strategy for conducting analysis on these datasets
  - FastBit: Bitmap index generation and querying software
  - Hybrid Parallel FastQuery
    - ✓ API to generate bitmap indexes
    - ✓ API to query indexed or data from different data formats (HDF5, NetCDF, and ADIOS-BP)
- Visualization strategy for examining the datasets
  - Query-driven visualization using VisIt

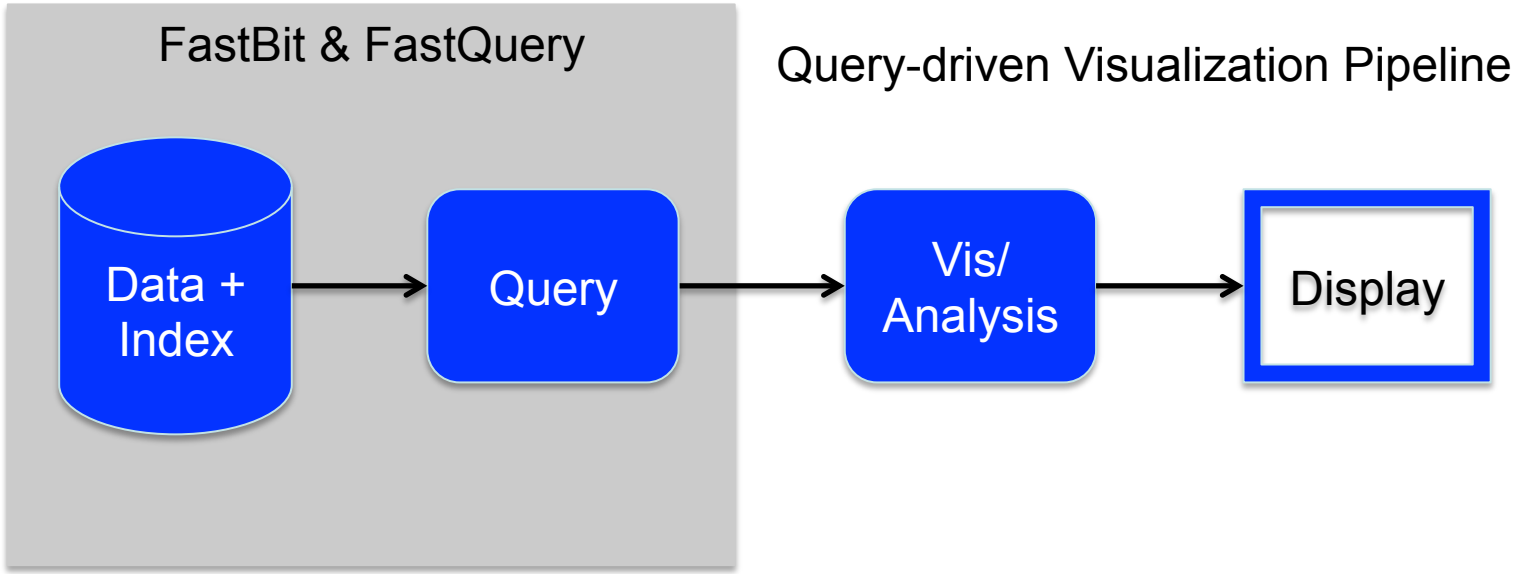
Visualization Challenge

- Visualizing too much data on computer displays is a challenge
  - Typical computer displays have O(1M) pixels and brute force rendering a trillion particles implies an overdraw factor of O(1M)

Traditional Visualization Pipeline

Query-driven Visualization

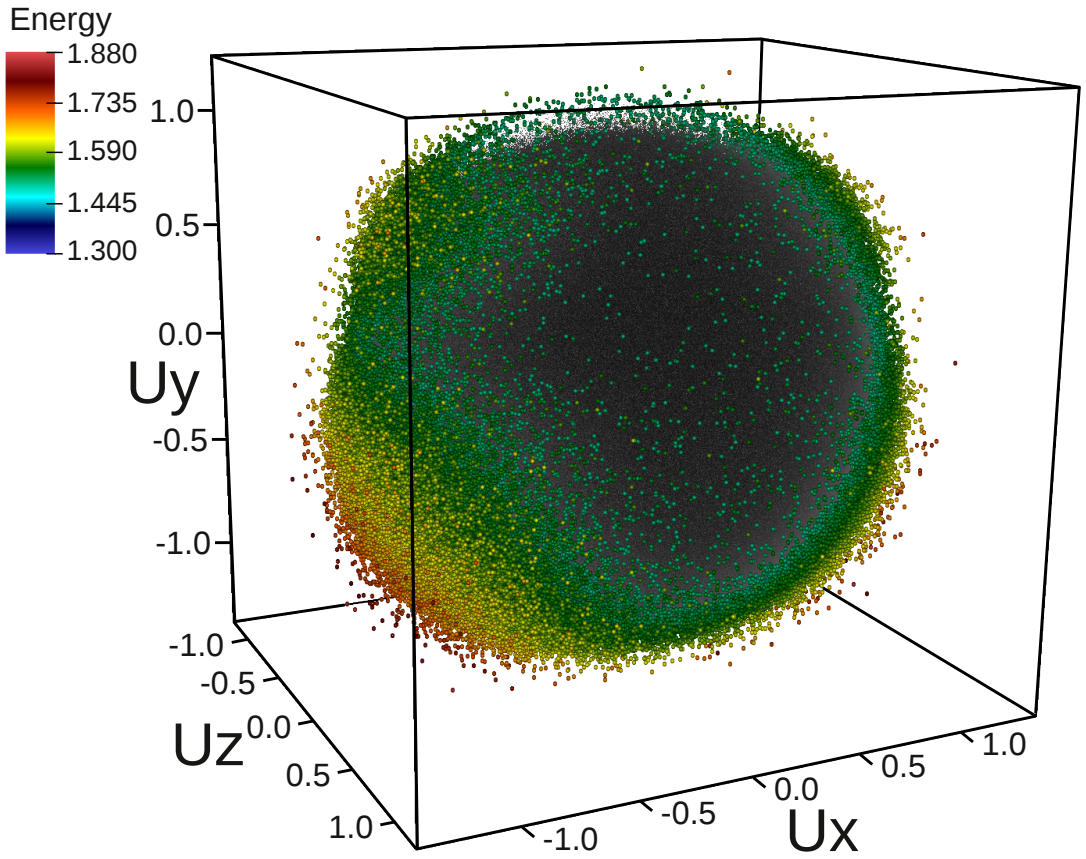
- Reduced the number of particles before rendering by down-selecting the scientifically interesting features
  - Highly energetic particles in this case
- Used Cross-Mesh Field Evaluation (CMFE)
  - Correlate particle data with the underlying field data



Science question

Are the highly energetic particles preferentially accelerated along the magnetic field?

- Data at time step 1905
- Showing all the particles with ‘Energy > 1.3’ in gray and those with ‘Energy > 1.5’ in color
- 165 million particles with Energy > 1.3 and 423,998 particles with Energy > 1.5

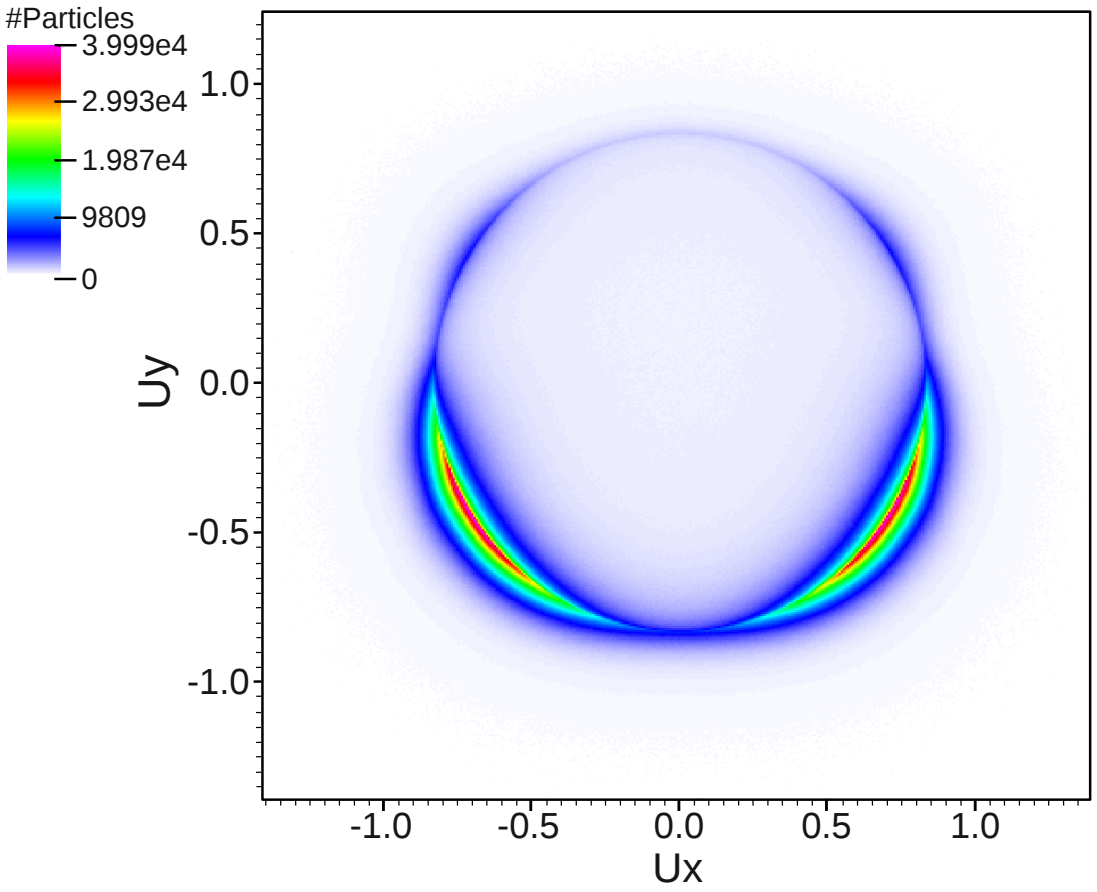


BERKELEY LAB

Science question

Are the highly energetic particles preferentially accelerated along the magnetic field?

- Data at time step 1905
- Showing the density of all particles with 'Energy > 1.3' in the U_x – U_y plane
- Shows preferential acceleration of the plasma in the direction parallel to the average magnetic field as evidenced by the highly distorted distribution function in the x-y plane

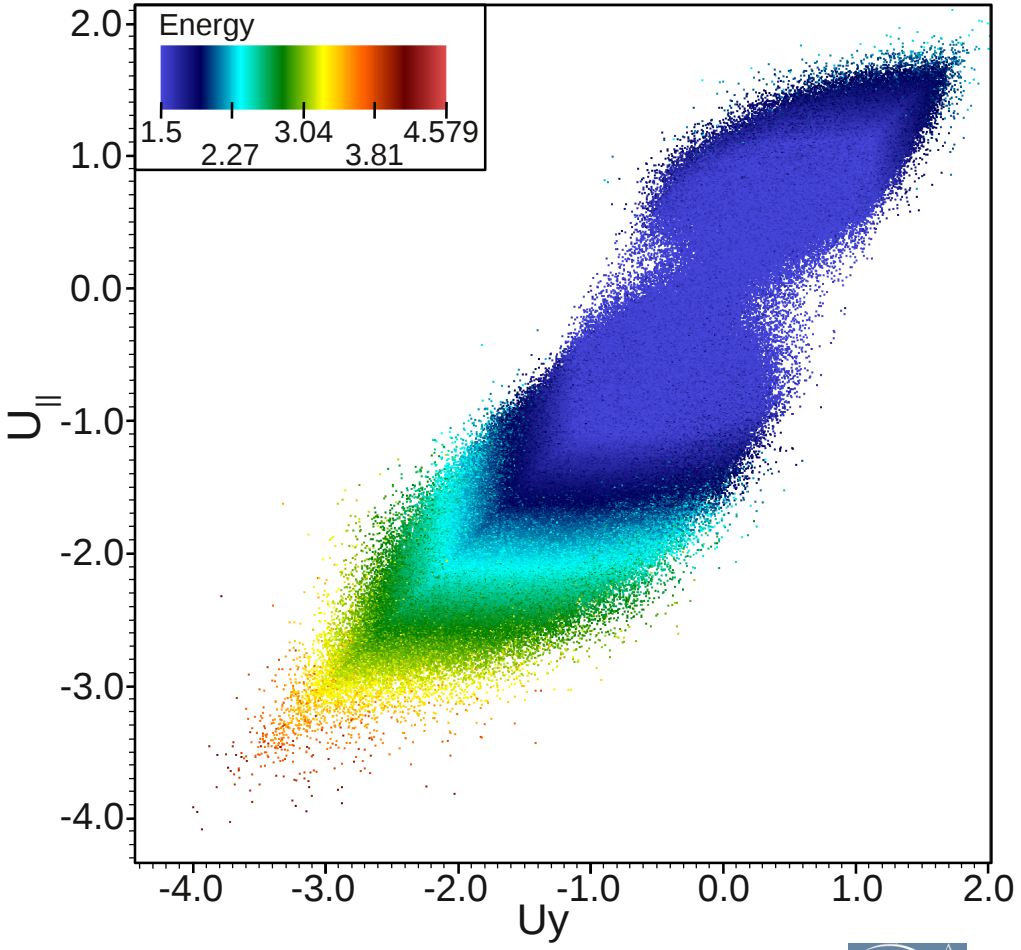


BERKELEY LAB

Science question

Are the highly energetic particles preferentially accelerated along the magnetic field?

- Scatter plot showing all particles with ‘Energy > 1.5’ in U_y and $U_{||}$ space colored by Energy
- Strong correlation between U_y and $U_{||}$
- The particles of high energy appear in regions of high negative $U_{||}$
- The highly energy particles are aligned (i.e., move parallel) to the magnetic field.

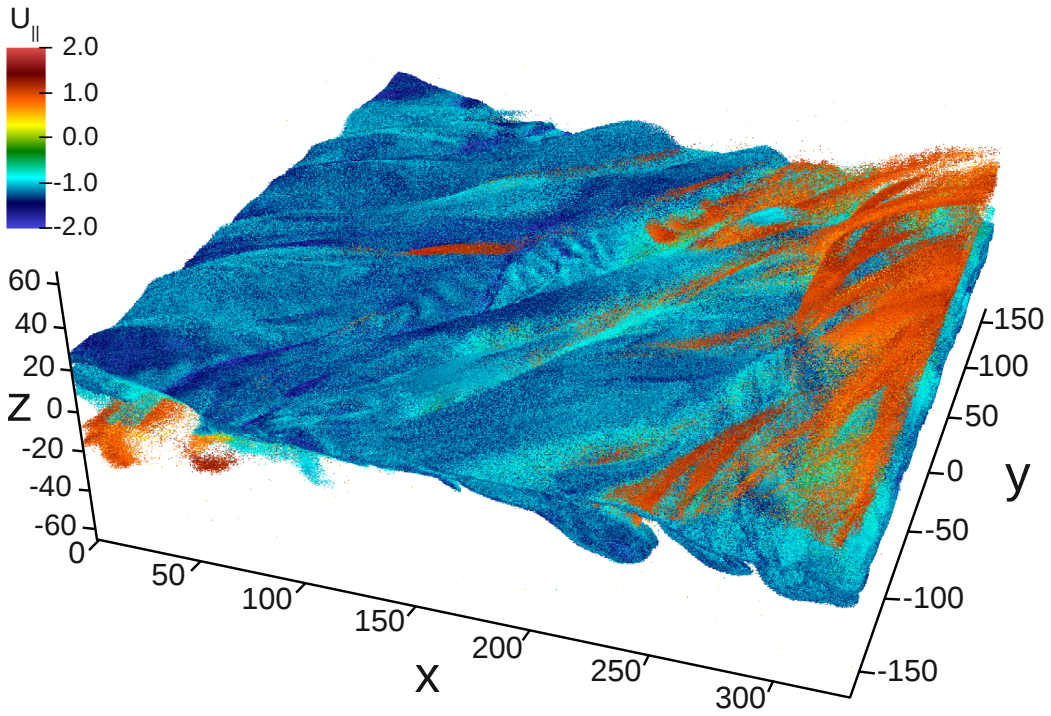


BERKELEY LAB

Science question

What is the spatial distribution of highly energetic particles?

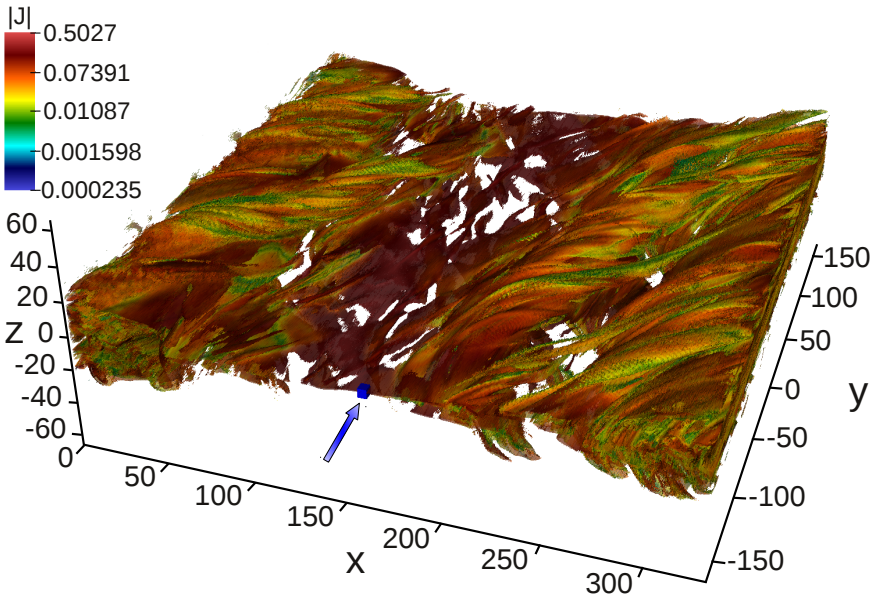
- Plot showing all particles with ‘Energy > 1.5.’
- The query selects ~0.05% of all particles
- Different particle structures with strong positive (red) and negative (blue) $U_{||}$ values

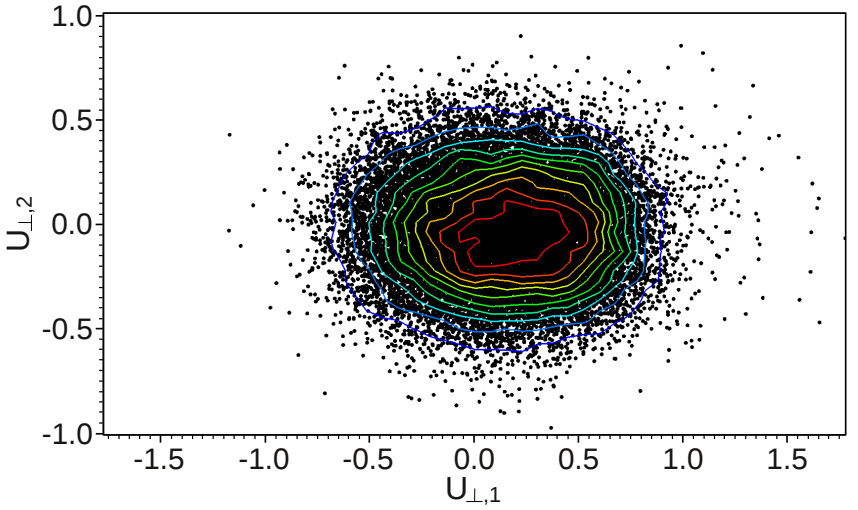


Science question

What are the properties of particles near the reconnection hot-spot?

- The X-line, where magnetic reconnection happens
- Particle distribution of $U_{\perp,1}$ $vs.$ $U_{\perp,2}$ in the vicinity of X-line
- The lack of cylindrical symmetry about the local magnetic field, called Agyrotropy
- Visualization of this data confirms the expected signature of the reconnection site in collisionless plasma





Wrapping up

- Addressed the data management and analysis challenges posed by a highly scalable plasma physics simulation
  - Storage: 24 to 27 GB/s (75% of peak parallel I/O BW)
  - Indexing: 9 minutes
  - Querying: ~3 seconds
- Demonstrated that we can handle big data challenges posed by exploratory analysis
- Facilitated new scientific discoveries
  - Application scientists explored and gained insights from the massive particle datasets for **the first time**
  - Capabilities developed here unlocked the scientific insights in unprecedented data

Thanks!

Co-authors:

J. Chou, O. Rübel, Prabhat, H. Karimabadi, W. S. Daughton, V. Roytershteynz, E. W. Bethel, M. Howison, K.-J. Hsu, K.-W. Lin, A. Shoshani, A. Uselton, and K. Wu

Advanced Scientific Computing Research (ASCR) for funding the ExaHDF5 Project; Program Manager: Lucy Nowell

Contact: Suren Byna
SByna@lbl.gov